

## PATENT ABSTRACTS OF JAPAN

(11)Publication number : 09-270793

(43)Date of publication of application : 14.10.1997

(51)Int.Cl.

H04L 12/18

G06F 13/00

H04L 29/08

H04N 7/14

(21)Application number : 08-081243

(71)Applicant : SONY CORP

(22)Date of filing : 03.04.1996

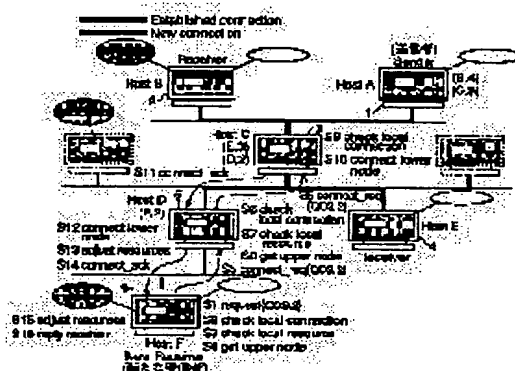
(72)Inventor : MURATA SEIJI  
SHIONOZAKI ATSUSHI

## (54) COMMUNICATION CONTROL METHOD

## (57)Abstract:

**PROBLEM TO BE SOLVED:** To efficiently perform multi-cast communication by integrating route control and a resource reservation mechanism.

**SOLUTION:** Corresponding to a connection establishment request with a transmitter (host A) from a new receiver (host F), resources are temporarily reserved, a route suitable for the request of the receiver is selected and the connection establishment request is transferred to a high-order node. A similar processing is executed in the host D as well and the connection establishment request is transferred to the high-order node. In the host C, the resources are reserved, a confirmation request is transferred to a low-order node and it is transferred through the host D to the host F. In the host F, the amount of the resources is adjusted and the establishment of connection with the host A is completed.



## LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平9-270793

(43) 公開日 平成9年(1997)10月14日

(51) Int.Cl. <sup>8</sup>	識別記号	庁内整理番号	F I	技術表示箇所
H 0 4 L 12/18		9466-5K	H 0 4 L 11/18	
G 0 6 F 13/00	3 5 5		G 0 6 F 13/00	3 5 5
H 0 4 L 29/08			H 0 4 N 7/14	
H 0 4 N 7/14			H 0 4 L 13/00	3 0 7 A

審査請求 未請求 請求項の数 4 O L (全 13 頁)

(21) 出願番号 特願平8-81243

(22) 出願日 平成8年(1996)4月3日

(71) 出願人 000002185

ソニー株式会社

東京都品川区北品川6丁目7番35号

(72) 発明者 村田 誠二

東京都品川区北品川6丁目7番35号 ソニー株式会社内

(72) 発明者 塩野崎 敦

東京都品川区北品川6丁目7番35号 ソニー株式会社内

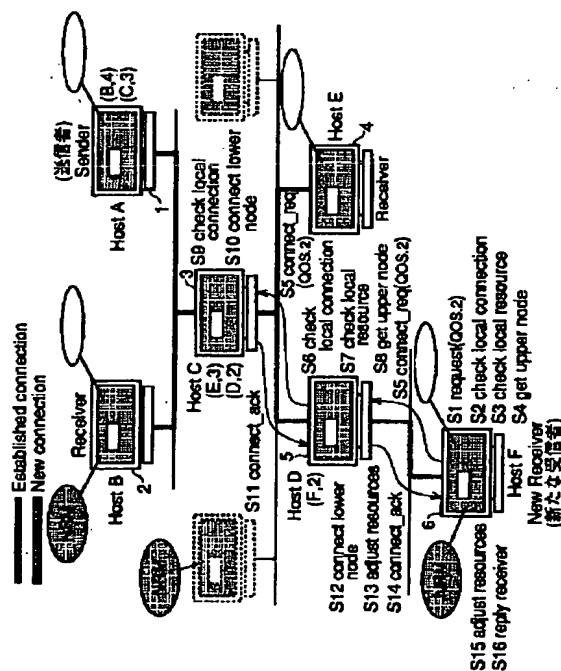
(74) 代理人 弁理士 稲本 義雄

(54) 【発明の名称】 通信制御方法

(57) 【要約】

【課題】 経路制御と資源予約機構を統合し、効率的にマルチキャスト通信を行うことができるようにする。

【解決手段】 新たな受信者（ホストF）からの送信者（ホストA）との接続設立要求に応じ、資源が仮予約されるとともに、受信者の要求に適した経路が選択され、接続設立要求が上位ノードに転送される。ホストDにおいても同様の処理が実行され、接続設立要求が上位ノードに転送される。ホストCにおいては、資源が予約され、確認要求が下位ノードに転送され、それがホストDを介してホストFに転送される。ホストFにおいては、資源の量の調整が行われ、ホストAとの間の接続の設立が完了する。



## 【特許請求の範囲】

【請求項1】 ネットワーク上の複数の送信者と複数の受信者との間にコネクションを設立し、前記送信者と前記受信者との間の通信を制御する通信制御方法において、

前記受信者からの要求に基づいて、前記送信者と前記受信者との間の経路を選択し、前記コネクションを設立する経路選択機構と、前記経路の資源を予約する資源予約機構の間のインタフェースを規定し、前記経路選択機構と前記資源予約機構とを統合することを特徴とする通信制御方法。

【請求項2】 前記経路選択機構が前記経路の選択を行う場合において、前記経路の資源についての評価が必要となき、前記資源予約機構は前記インタフェースを介して前記経路の資源を評価し、前記経路選択機構は前記評価結果に基づいて前記経路の選択を行うことを特徴とする請求項1に記載の通信制御方法。

【請求項3】 ネットワーク上の複数の送信者と複数の受信者の間にコネクションを設立し、前記送信者から前記受信者へのデータの転送を制御する通信制御方法において、

各送信者が保持する各送信者のリストに基づいて、前記受信者から各送信者に対して前記コネクションの設立を行い、  
複数の前記送信者が前記コネクションを共有する場合、前記コネクションに予約された資源の量に基づいて、前記コネクションを介して前記データを前記受信者に同時に送信する前記送信者の数を所定数以下に制限し、  
新たな送信者または受信者の前記ネットワークへの参加要求は、前記新たな送信者または受信者が前記ネットワークに参加した場合においても、既存の前記コネクションを設立している前記送信者および受信者のQOS要求が保証されるとき、受理されることを特徴とする通信制御方法。

【請求項4】 前記受信者は、前記送信者の所定のものからの前記データだけを受信することを特徴とする請求項3に記載の通信制御方法。

## 【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は、通信制御方法に関し、例えばマルチキャスト通信等に用いて好適な通信制御方法に関する。

【0002】

【従来の技術】 近年、計算機がネットワークによって相互接続された分散環境において、計算機を利用した電子会議やビデオオンデマンド（Video on demand）システムなどのマルチメディアアプリケーションが注目を浴びている。このようなアプリケーションが行う通信の特徴として、データ転送に実時間性制約、即ち、一定レート、または一定時間内の配送などの制約が存在すること

と、同時に同一のデータを、複数の受信者に対して効率良く転送する必要があることなどが挙げられる。

【0003】 マルチキャストに対応した資源予約を行うプロトコルは既にいくつか提案されている。ここで、資源とは、CPU（Central Processing Unit）処理時間、ネットワーク帯域、およびデータバッファ等である。例えば、ST-11（Internet Stream Protocol, Version 2）は、1対多のコネクションを設立するプロトコルである。送信者は最適なFlowSpec（Flow Specification）を提案し、全受信者に対してコネクション設立要求をマルチキャストする。ここで、FlowSpecとは、データ転送レート、および許容伝送遅延などのデータ転送の性質を表し、データ転送に必要とされる資源の量は、指定されたFlowSpecをもとに計算される。

【0004】 各受信者は、送信者からの経路で保証可能な最大FlowSpecを送信者に送り返す。送信者は、全受信者から集めたFlowSpecの中で、最低レベルのものを選択し、再びコネクション設立要求を送信する。2度目のコネクション設立要求において、実際に資源が予約され、全受信者に対して、等しいFlowSpecのコネクションが設立される。このコネクション設立手続きは、受信者が参加または脱退する度に行われる。

【0005】 また、RSVP（Resource ReSerVation Protocol）においては、予め通常のマルチキャストを用いて定期的にツリー構築要求を受信者に送ること、RSVPが管理するマルチキャストツリーを構築しておく。新たに参加する受信者は、構築されたツリーを遡って、コネクション設立要求を送ることにより、受信者からコネクションを設立する。この方式では、コネクション設立は受信者毎に行われるため、各受信者毎に異なるFlowSpecを設定することが可能である。RSVPでは、設立されたコネクションは静的ではなく、定期的に送られてくるツリー構築要求に従い、動的に再設立される。

【0006】

【発明が解決しようとする課題】 しかしながら、従来のネットワークプロトコルでは、データ転送時に利用可能な資源の量に応じて、データ転送に要する時間が変化するため、データ転送の実時間性を満たすことは困難である課題があった。従って、送信を始める前に、送信者と受信者間にコネクションを設立し、転送要求に応じて必要な資源を予め予約するプロトコルが必要とされる。

【0007】 特に、マルチキャスト通信（1対多通信）では、受信者が複数であり、受信者の参加または脱退により、データ転送中にコネクションが動的に変化する。そこで、送信者と受信者間のコネクションをツリー上に設立することにより、経路を共有し、かつ受信者の要求に応じて動的にコネクション設立が可能な実時間マルチ

キャストプロトコルが求められている。

【0008】上記2つのプロトコルST-11とRSVPの特徴をまとめたものを図11に示す。図11に示したように、ST-11は、受信者参加時に、毎回送信者から資源予約を繰り返すため、受信者参加時のコネクション設立要求の付加が送信者に集中し、グループのメンバーサイズのスケラビリティに問題があり、また、各受信者の転送レベルが最も低い受信者に合わされてしまうという課題があった。

【0009】そこで、RSVPにおいては、受信者からコネクション設立要求を送信することにより、上記課題を解決している。しかしながら、経路が動的に変化することから、既設コネクションの保証が十分ではなく、また、受信者毎の要求に対応しようとしているが、経路選択自体は送信者によって行われるため、各受信者の要求に応じた最適な経路を選ぶことができない課題があった。

【0010】このように、従来のプロトコルの問題点は、経路制御と資源予約機構を全く独立した機構として捉えているため、経路選択を送信者からしか行うことができないことから生じていると考えられる。しかし、既存のマルチキャスト経路制御アルゴリズムでは、内部の処理では受信者側から、送信者側への経路選択が可能である。

【0011】本発明はこのような状況に鑑みてなされたものであり、経路制御機構と資源予約機構間のインタフェースを定め、統合することにより、各受信者からの要求に応じた経路制御、およびコネクションの設立を可能とするものである。

【0012】

【課題を解決するための手段】請求項1に記載の通信制御方法は、受信者からの要求に基づいて、送信者と受信者との間の経路を選択し、コネクションを設立する経路選択機構と、経路の資源を予約する資源予約機構の間のインタフェースを規定し、経路選択機構と資源予約機構とを統合することを特徴とする。

【0013】請求項3に記載の通信制御方法は、各送信者が保持する各送信者のリストに基づいて、受信者から各送信者に対してコネクションの設立を行い、複数の送信者がコネクションを共有する場合、コネクションに予約された資源の量に基づいて、コネクションを介してデータを受信者に同時に送信する送信者の数を所定数以下に制限し、新たな送信者または受信者のネットワークへの参加要求は、新たな送信者または受信者がネットワークに参加した場合においても、既存のコネクションを設立している送信者および受信者のQOS要求が保証されるとき、受理されることを特徴とする。

【0014】請求項1に記載の通信制御方法において、受信者からの要求に基づいて、送信者と受信者との間の経路を選択し、コネクションを設立する経路選択機

構と、経路の資源を予約する資源予約機構の間のインタフェースを規定し、経路選択機構と資源予約機構とを統合する。

【0015】請求項3に記載の通信制御方法においては、各送信者が保持する各送信者のリストに基づいて、受信者から各送信者に対してコネクションの設立を行い、複数の送信者がコネクションを共有する場合、コネクションに予約された資源の量に基づいて、コネクションを介してデータを受信者に同時に送信する送信者の数を所定数以下に制限し、新たな送信者または受信者のネットワークへの参加要求は、新たな送信者または受信者がネットワークに参加した場合においても、既存のコネクションを設立している送信者および受信者のQOS要求が保証されるとき、受理される。

【0016】

【発明の実施の形態】以下、本発明を適用した実時間マルチキャストプロトコルRtMP (Real-time Multicast Protocol) の詳細について説明する。

【0017】汎用性を維持するためには、経路制御と資源予約を統合する際に、特定の経路制御アルゴリズムに依存してはならない。そこで、経路制御に関する機構を、経路制御アルゴリズムに依存する部分と、それとは独立の部分に分割する。

【0018】例えば、経路制御アルゴリズムに依存する部分は、ノード間での経路情報の交換、および経路計算を行うこととし、経路制御アルゴリズムに独立な部分は、経路情報の資源に関する部分の管理および評価、提案された経路からの所定の経路の選択、そして既設のコネクションに関する情報を保持することとする。

【0019】経路を選択する際に、資源についての評価が必要な場合には、図1に示したようなインタフェースの内の3つ(RouteAdd、RouteSelect、およびRouteEval)を用い、資源予約モジュール側で評価を行う。コネクションを設立する際には、資源予約モジュールはインタフェースRouteReqを用いて上位ノードの候補を取得し、その中から受信者の要求を満たすのに最適な上位ノードを選択する。

【0020】この機構により、特定の経路制御アルゴリズムに依存することなく、資源に関するパラメータを経路情報に反映させることができ、受信者の要求に応じた経路選択が可能となる。

【0021】様々な受信者の要求に対応するためには、各経路毎にその経路が満足できるFlowspecのコネクションを設立することが可能でなくてはならない。すなわち、各経路毎に異なるFlowspecのコネクションを設立することが可能でなければならない。

【0022】つまり、送信者が送出したデータを全経路に転送できるとは限らず、図2に示したように、経路が分かれる場合には、各経路の転送能力を確認し、転送可能な経路にのみデータを送る必要がある。その際に、転

送能力を越える量のデータを転送してしまった場合には、資源が余分に使用され、そのコネクションの実時間性を保証することができなくなるだけではなく、他のコネクションにも悪影響を及ぼす可能性がある。

【0023】そこで、送信データに優先度レベルを設け、そのレベルに応じて、中間ノードはデータを転送する経路を選択するようにする。RtMP内部では、優先度レベルはデータ転送レートにマップされる。図2に示した中間ノードの場合、転送レートが512Kbps（キロバイト/秒）の経路と転送レートが1Mbps（メガバイト/秒）の経路に分岐している。また、転送レートが512Kbpsの経路に転送可能なデータは優先度レベルが2以下のものとされ、転送レートが1Mbpsの経路に転送可能なデータは優先度レベルが3以下のものとされている。

【0024】図3に、優先度レベルとデータ転送レートとの対応関係を示す。優先度レベルとデータ転送レートとの対応は、送信者によって指定される。受信者は、コネクション設立を要求する際に、Flowspecのパラメータとして、受信すべき優先度レベルを指定する。

【0025】送信ノードにおいて、データは優先度レベルで指定されたデータ転送レートを越えないようにレート制御される。各中間ノード、および受信ノードでは、コネクション設立時に指定された優先度レベルに従い、転送されてきたデータを転送し、または受信する。

【0026】このデータ転送制御機構を送信者側から見た場合、送信者は多重化されたチャネルを有し、各データの重要度に応じたデータ送信を行うことが可能であるということになる。従って、例えば、図3に示したように、優先度レベルが1の場合、音声を送信し、優先度レベルが2の場合、粗い画像を送信する。また、優先度レベルが3の場合、細かい画像を送信する。これにより、資源に余裕があるノードでは細かい映像、資源が少ないノードでは、粗い画像または音声のみでサービスの提供を受けることが可能となる。

【0027】優先度レベル以外に、到着時間制約（一定時間以内にデータが届くという制約）等の情報もコネクション情報として存在し、優先度レベルと同様に中間ノードで管理され、コネクション設立時、データ転送時に参照される。

【0028】図4に、コネクション設立手順の例を示す。ここでは、送信者であるホスト1（=HostA）と受信者であるホスト2（=HostB）、ホスト4（=HostE）間に張られているコネクションに、新たな受信者であるホスト6（=HostF）が参加する場合に行われる処理が示されている。図4において、矢印はメッセージの流れを示し、ホスト名の下に記載されている（下位ノード、優先度レベル）はコネクション情報を示している。

【0029】ステップS1乃至S3において、受信者か

らのコネクション設立要求（request（QOS，2）（この場合、QOS（Quality Of Service）が2に対応するデータの送信を要求している））に応じ、資源予約モジュールが資源の仮予約を行う。

【0030】ステップS4においては、経路制御モジュールに上位ノードを問い合わせ、受信者の要求に適した経路を選択する（get upper node）。次に、ステップS5において、コネクション設立要求（connect\_req（QOS，2））を上位ノード（この場合、ホスト5（=ホストD））に転送する。

【0031】ホストDは、ホストFからのこのコネクション設立要求を受信したとき、上位ノード（この場合、ホスト3（=ホストC））との間にコネクションを設立しておらず、受信者の要求を満たすことができないので、ホストFにおけるステップS2乃至S4の処理に対応するステップS6乃至S8の処理を実行し、資源の仮予約が行われるとともに、コネクション設立要求がさらに上位のノードに転送される。

【0032】ホストDからのこのコネクション設立要求を受信したホストCは、ステップS9において、既にホストAとの間にコネクションを設立していることを認識し、この場合、受信者の要求を満たすこと（所定のFlowspecを満たすこと）が可能であるので、資源の予約を行い、ステップS10において下位ノードとのコネクションを設立し、ステップS11において、コネクション情報に加え、確認要求（Connect\_ack）を下位ノードに送り返す。

【0033】一方、設立されているコネクションが要求を満たさない場合、Flowspecの変更が試みられる。このFlowspecの変更処理は、上述したステップS1乃至S3において行われる資源予約要求処理と基本的には同様の手順で行われるが、ここでのFlowspecの変更処理は、既設コネクション上で行われる点、および新たに資源を確保するのではなく、現在の資源を変更する点で上述した資源予約要求処理とは異なる。

【0034】上位ノードとの間のコネクションが設立されていない場合、あるいは上述したFlowspecの変更が不可能である場合、上述した場合と同様にして、資源予約要求処理が繰り返される。

【0035】上位ノードからの確認要求を受け取ったノードは、上位ノードで保証されたFlowspecに従って、先に仮予約をした資源の量を調整し、下位ノードに確認要求を送る。この場合、ホストCからの確認要求を受け取ったホストDは、ステップS12において、下位ノードとの間のコネクションを設立するとともに、ステップS13において、先に仮予約を行った資源の調整を行い、ステップS14において、下位ノードに対して確認要求を送信する。

【0036】このようにして、上位ノード（この場合、

ホストC)からの確認要求が受信者のノード(この場合、ホストF)に戻ると、ステップS15において、資源の量の調整が行われ、その時点で、コネクション設立が完了する。次に、ステップS16において、コネクション設立が完了したことが受信者(ホストF)に対して通知される。

【0037】この方式では、受信者の要求に応じて経路が選択されるため、要求に最適なコネクションの設立が可能となる。また、コネクション設立処理は、受信者と接続されるノード間のみに于行われるため、グループのスケラビリティは維持される。さらに、各ノードにおいて資源の残量を確認することにより、既設のコネクションを乱さないことを保証することが可能である。

【0038】RtMPにおいては、設立されたコネクションは、受信者の参加または脱退がない限り通常変化しない。これは設立されたコネクションは障害発生時以外には常にデータ転送を保証する(プロトコルの動作によってコネクションが乱されることはない)点を実時間通信プロトコルの必要条件としているためである。

【0039】しかしながら、コネクションが静的な場合、ホストやネットワークの障害による経路変更がコネクションに反映されないため、明示的に障害回復処理を行う必要がある。障害が生じた場合の処理を以下に示す。

【0040】図5は、障害が発生した場合の処理例を説明するためのフローチャートである。まず、ステップS31において、経路制御部は、隣接ノードに定期的に「HELLOメッセージ」の送信を行う。ステップS32においては、隣接ノードからの応答があったか否かが判定される。隣接ノードからの応答があったと判定された場合、障害の発生はないものとし、ステップS31に戻り、再度ステップS31以降の処理が実行される。一方、隣接ノードからの応答がなかったと判定された場合、隣接ノードまたは隣接ノードとの間のネットワークに障害があったものとみなし、ステップS33に進み、障害発生時の処理を開始する。

【0041】ステップS33においては、通信不能となったノードとの間に、コネクションが存在していたか否かが判定される。コネクションが存在していなかったと判定された場合、処理を終了する。一方、コネクションが存在していたと判定された場合、ステップS34に進み、そのコネクションの破棄を行う。

【0042】次に、ステップS35において、通信不能となったノードが下位ノードであるか否かが判定される。通信不能となったノードが下位ノードであると判定された場合、下位ノードに対してデータを送信する必要がなくなるため、ステップS36に進み、下位ノードの資源が解放される。

【0043】次に、ステップS37において、このと

き、下位ノードがただ1つであるか否かが判定される。下位ノードが1つであると判定された場合、上位ノードにコネクション破棄要求を送り、上位ノードの資源も解放し、処理を終了する。

【0044】一方、ステップS35において、下位ノードとの通信が不能ではないと判定された場合、即ち、上位ノードと通信不能となったと判定された場合、ステップS39に進み、他の経路を検索する。

【0045】次に、ステップS40において、他の経路が存在するか否かが判定され、他の経路が存在すると判定された場合、ステップS41に進み、コネクション設立要求を発行し、他の経路とのコネクションを設立する。一方、他の経路が存在しないと判定された場合、ステップS42に進み、上位ノードの資源を解放し、下位ノードにコネクション再設立要求メッセージを送る。

【0046】ステップS43においては、コネクション再設立要求を受け取ったノードは、現在のコネクションを破棄した後、他の経路を検索し、コネクションの張り直しを試みる。次に、ステップS44に進み、受信者への通知を行う。即ち、コネクションを破棄すると同時に、下位ノードに存在する受信者に対して、障害が発生したことを通知し、さらに、コネクションの張り直しが完了した場合、そのことを再び下位ノードに存在する受信者に通知する。そして、処理を終了する。

【0047】このような処理によって、受信者は、障害が発生した場合において、データを転送するとき、実時間性を守ることができないことを知ることができ、その対応処置を行うことができる。また、可能な限りコネクションは再び設立される。これらは、障害対処の重要な機能である。

【0048】例えば、電子会議のように、1対多通信だけでなく、多対多通信を必要とするアプリケーションも多数存在する。特に、いくつかのアプリケーションは、同時には固定数の送信者しかデータを送信しないと特徴を有する。この場合、各送信者についてコネクションを設立するのではなく、通信の性質を考慮し、コネクションを共有することにより、効率的に資源を利用することが可能である。

【0049】多対多コネクションを効率良くサポートするために必要な機構を以下にまとめる。

【0050】まず、コネクションの共有を行う機構が必要である。共通の経路では、最大送信者数に対応する帯域を予約すればよい。ただし、送信者間で協調し、予約された送信者数以上の送信者が同時にデータを送信しないことを保証するプロトコルを必要とする。

【0051】例えば、次のようなプロトコルを用いることにより、受信者に対して同時にデータを送信する送信者数を所定数以下に制限することができる。

【0052】即ち、全ての送信者は、送信者リストを持っているので、送信可能な送信者のリストも同様に管理

するようにする。全ての送信者は、送信者リストを持っているので、送信者同士でトークンをやりとりし、トークンを渡された送信者は、受信者に送信すべきデータがある場合、それを受信者に送信し、所定のタイミングでトークンを次の送信者に渡す。また、送信すべきデータがない場合、データの送信は行わず、所定のタイミングでトークンを次の送信者に送信する。

【0053】ここで、トークンは、例えば、送信者リストに記述される各送信者に設定された優先度の順番に従って、次の送信者に送信されるようにすることができる。各送信者がトークンを保持する時間も、この優先度に基づいて決めることができる。これにより、受信者に同時にデータを送信する送信者の数を1に制限することができる。

【0054】上記プロトコルを応用し、2以上の所定数のトークンを送信者間でやりとりさせるようにすることにより、受信者に同時にデータを送信することができる送信者数を2以上の所定数以下に制限することができる。従って、上記機構を用いることにより、受信者にデータを同時に送信可能な送信者の数を、1または2以上の所定数以下に制限することができる。

【0055】次に、受信チャネルの選択を行う機構が必要である。各受信者は、全送信者からのデータを受け取るのではなく、ある特定の送信者からのデータだけを受け取るようにする。

【0056】このとき、送信者同士が協調するか、あるいは受信者が任意の送信者に対してコネクションの設立を行うために、送信者のリストが必要となる。RtMPにおいては、送信者リストを管理するプロトコルを新たに加えることにより対応する。以下、RtMP内で追加、変更される部分の詳細について説明する。

【0057】まず、送信者リストの管理について説明する。送信者リストを管理する代表者を決め、その代表者に送信者リストを集中管理させることは容易であるが、送信者リストの更新、および保持が1点に集中するため、管理者に障害が発生した場合、プロトコル動作に致命的な障害が生じる。また、更新負荷が集中するなどの問題も生じる。

【0058】そこで、RtMPにおいては、送信者リストを各送信者毎にそれぞれ保持する方法を採る。送信者リストの一貫性を保証するために、以前、本発明者が提案したProcess Group Management Protocol (PGMP)において、グループメンバリスト管理に用いられている、更新に全順序関係を与える方式を採る。例えば、複数の送信者が送信者リストの更新を同時に行おうとした場合、古い送信者リストに基づいて、送信者リストが更新される可能性がある。これを防ぐために、更新要求に全順序関係を付け、全ての送信者が同一の順序で更新要求を受け取り、古い送信者リストに基づいた更新要求を拒否するようにす

る。

【0059】即ち、所定の送信者（以下、送信者Aとする）が各送信者に送信した更新要求を各送信者が受信すると、その更新要求に、まだ実行が不可能であることを示す所定の実行不可能の印を対応付け、更新要求待ち行列に入れる。そして、各送信者は確認応答を、更新要求を受信した順番を示すシーケンシャル番号とともに送信者Aに送り返す。

【0060】送信者Aが全ての送信者から確認応答とシーケンシャル番号を受け取ったとき、2度目の更新要求を、受け取ったシーケンシャル番号の中で最も大きいものとともに、各送信者に送信する。各送信者は、送信者Aからの2度目の更新要求とシーケンシャル番号を受け取ると、待ち行列の中にある最初に受信した更新要求のシーケンシャル番号を、いま送信者Aより送信されてきたシーケンシャル番号に変更し、実行不可能の印を実行可能の印に変え、シーケンシャル番号で昇順にソートする。ここで、待ち行列の中にある複数の更新要求に同一のシーケンシャル番号が対応するような場合、所定の統一的な基準、例えば、送信者（ホスト）のアドレス等によって、その順序づけを行うようにする。

【0061】このソートによって先頭に並び変えられた更新要求に、実行可能の印が対応付けられている場合、その更新要求が現在の送信者リストに基づいたものであるかを、送信者リストに対応付けられたバージョンを比較することにより確認する。確認の結果、現在のバージョンである場合、送信者Aからの更新要求が受け入れられ、各送信者は各送信者が保持する送信者リストを更新し、送信者リストのバージョンを1だけ増加させ、更新完了通知を送信者Aに送信する。

【0062】一方、確認の結果、更新要求が現在の送信者リストに基づいたものではなく、古い送信者リストに基づいたものである場合、各送信者は更新要求を拒否し、拒否メッセージを送信者Aに送信する。送信者Aが各送信者より拒否メッセージを受け取った場合、再び各送信者に対して更新要求を行い、上述した処理が繰り返される。

【0063】このような機構により、各送信者は常に、現在、グループにデータを送信する可能性のある送信者リストを保持することが可能となる。

【0064】新たな送信者が参加した場合、各受信者に対し、コネクションを設立する必要がある。RtMPにおいては、通常のコネクション設立は、受信者参加時において受信者から行われるため、特別なコネクション設立手続きが必要とされる。RtMPにおける送信者参加手続きを、図6に示したフローチャートを参照して説明する。

【0065】まず、ステップS51において、送信者リストの取得および更新を行う。即ち、送信者リストを他の送信者から取得し、他の送信者が保持するリストに自



ホストを加える。

【0066】次に、ステップS52に進み、受信者に対し、コネクション設立を要求する。この受信者に対するコネクション設立要求には、まだ、受信者に対してコネクションが張られていないため、経路制御部が提供する通常のマルチキャストが用いられる。各中間ノードにおいてこの要求を転送した下位ノードを記録しておき、確認応答の集計に備える。

【0067】次に、ステップS53において、コネクションの設立を必要とするか否かが判定される。即ち、この要求を受けた受信者が、全送信者からの受信を望んでいた場合、新たな送信者との間のコネクションの設立が必要であるとし、ステップS54に進む。

【0068】ステップS54においては、受信者は、新しい送信者に対して、コネクションの設立を行う。ここでのコネクションの設立は、ST-II (Internet Stream Protocol, Version 2) のように、送信者からの要求メッセージにより構成されたツリーに沿って行われるのではなく、受信者の要求を考慮して通常のコネクション設立と同様にして行われる。

【0069】次に、ステップS55に進み、コネクションの設立が完了した場合、コネクション設立完了の確認応答を送信者に対して送信する。

【0070】一方、ステップS53において、コネクションの設立を必要としないと判定された場合、即ち、受信者が送信者とのコネクションの設立を拒否した場合、ステップS56に進み、拒否メッセージが送信者に送り返される。

【0071】ステップS55における確認応答およびステップS56における拒否メッセージは、送信者からのコネクション設立要求が受信者に転送されてきた時に構成されたツリーに沿って、送信者に送られる。

【0072】次に、ステップS57において、コネクション設立の確認を行う。即ち、送信者は、全下位ノードからの確認応答または拒否メッセージが来るのを待ち、全下位ノードからの確認応答または拒否メッセージが来た時点で、送信者参加手続きを終了する。

【0073】ここで、拒否メッセージが来た場合には、送信者はグループへの参加を許されない。この場合、既に行われた送信者参加手続きの取り消し処理を行う。

【0074】この機構により、送信者が新たに参加した場合、既存の受信者に対して自動的にコネクションが設立され、受信者は新しい送信者からのデータを受け取ることが可能となる。

【0075】次に、受信者が参加する場合の処理について、図7のフローチャートを参照して説明する。1対多の場合と比較して異なる点は、送信者が複数存在する可能性があるため、各送信者についてコネクション設立を行う必要がある点である。この際、コネクションの共有可能性などのコネクションの性質も考慮する必要があ

る。

【0076】まず、ステップS61において、受信者は送信者グループに対してリストを要求する。次に、ステップS62において、リストを得た受信者は、各送信者についてコネクション設立処理を行う。この際、既に設立されているコネクションと共有可能な場合は、送信者が異なるときにも、それ以上の資源予約を行わない。ただし、コネクション設立要求は上位ノードに送られる。これは、送信者が異なる場合、既設コネクションが対象送信者まで張られているとは限らないためである。

【0077】また、受信者が脱退する場合にも、1対多通信のときのように、単純にコネクションを破棄するのではなく、それが共有されているか否かの確認を行い、共有されている場合には解放しないようにする。コネクション設立時の場合と同様に、破棄要求自体は上位ノードに転送される。

【0078】なお、新たな送信者または受信者の参加要求は、既存のコネクションを設立している送信者および受信者のQOS要求を必ず保証するという条件で受理される。

【0079】次に、RtMPの実装について説明する。実時間オペレーティングシステムであるRT-Mach (Real-Time Mach) マイクロカーネル上で動作するRTS/NPS (Real-Time Server/Network Protocol Server) 環境上で実装を行った。ここでは、NPS内の1プロトコルとしてRtMPを実装し、同時に協調して経路制御を行うサーバ、および各ネットワークセグメントのネットワーク資源を管理するサーバの実装も行った。

【0080】図8は、RtMPの内部構造の例を示している。RtMPは、上述したように実時間オペレーティングシステムであるRT-Mach Micro Kernel 2.3上で動作するネットワークプロトコルサーバ (Network Protocol Server) 20内の1つのプロトコルとして実装され、ネットワークプロトコルサーバ20は、ネットワークデバイスレイヤ15、プロトコルレイヤ16、およびインタフェースレイヤ17より構成されている。ネットワークプロトコルサーバ20と経路制御を行うルーティングサーバ (Routing Server) 21は、所定のインタフェース処理等を行うコントロールスレッド (Control thread) 14を介して情報のやりとりを行うようになっている。

【0081】データ転送の実時間性は、設立されたコネクション毎に、input/output処理を行う周期的実時間スレッド (output thread、local send thread、およびforwarder thread) と専用バッファを割り当てることにより保証する。コネクション設立要求などの制御メッセージは実時間性を保証する必要がないため、NP S汎用のコントロールスレッド14によって処理され

る。

【0082】次に、各メッセージがどのように処理されるかについて説明する。ネットワークデバイス層（Network device layer）15のネットワークデバイスドライバによって受け取られたパケットは、まず、プロトコルレイヤ（Protocol layer）16のデータリンクプロトコルに割り当てられている高い優先度の「input thread」（図8においては、ether input thread 19）によって処理される。ここでは、etherのパケットヘッダの「protocol type」に従って、上位プロトコルの入力処理ルーチン（RtMP入力処理ルーチン）が呼び出される。

【0083】次に、RtMP入力処理ルーチンにより、RtMPパケットヘッダの「type field」の内容に基づいて、それがユーザデータおよび制御メッセージのいずれであるかが判別され、適切なスレッドにディスパッチされる。

【0084】ユーザデータであると判定された場合、宛先グループアドレスから接続テーブル（RtMP\_PORT 18）が検索される。RtMP\_PORT 18の経路情報部に従って、各接続毎に、ローカルに存在する受信者に転送するスレッド（図8においては、local send thread 12）と下位ノードに転送するためのスレッド（図8においては、forwarder thread 13）にデータが引き渡される。local send thread 12は、データを各受信者の受信バッファに入れる処理を担当し、forwarder thread 13は、RtMP\_PORT 18に示される下位ノードに対するデータ転送を行う。

【0085】一方、制御メッセージであると判別された場合、宛先グループアドレスからRtMP\_PORT 18が検索される。接続の状態と、各メッセージタイプに従って処理が行われ、必要に応じて、Routing-Server 21、図示せぬNetwork Resource Manager、その他のノードに転送される。

【0086】このように、プロトコル制御を行うスレッドとデータ転送を行うスレッドを分離し、専用のバッファを割り当てることにより、データ転送が他の処理によって妨げられることを抑制し、要求されたFlowspec通りのデータ転送を保証することができる。

【0087】次に、上記RtMPの性能評価、既存のプロトコルとの比較を行い、RtMPの有効性について検討する。

【0088】まず、性能評価を行う。専用Ethernetで接続されたPC-AT互換機（CPU：Pentium 90MHz）上で、RtMPの各プリミティブについての性能評価を行った。

【0089】図9は、他のメンバが同一のセグメントにいる場合と隣接セグメントにいる場合とについて、受信者、送信者の参加に要する時間を示している。上段は送信者のみが存在するグループに受信者が参加する場合、下段は送信者が1、受信者が1のグループ（送信者と受信者は同一のセグメントに存在する）に新たに送信者が加わる場合について測定した結果である。

【0090】この結果から、接続ノードおよび送信者までの距離とに影響されることがわかる。また、受信者数が増加した場合も接続ノードの増加はなく、むしろ既設接続との距離が短くなることにより、設定時間が短縮されたことがわかる。

【0091】次に、RtMPのプロトコルオーバーヘッド（パケットヘッダの構築および解析、接続情報の検索、接続に従ったデータ転送等に要する時間など）を反映し、アプリケーションの性能に最も影響するメッセージ送信のラウンドトリップ時間をメッセージサイズについて測定した結果について説明する。

【0092】図10は、その測定結果を表したグラフである。縦軸はラウンドトリップ時間（単位はミリ秒（msec））を表しており、横軸はメッセージサイズ（単位はバイト（bytes））を表している。点線で表したグラフは、他のメンバが同一セグメントにいる場合におけるUDP/IP（User Datagram Protocol/Internet Protocol）の往復通信遅延時間（round trip time）を表し、1点鎖線で表したグラフは、同様に、他のメンバが同一セグメントにいる場合におけるRtMPによる往復通信遅延時間を表している。さらに、実線で表したグラフは、他のメンバが隣接セグメントにいる場合におけるRtMPによる往復通信遅延時間を表している。

【0093】上記グラフから分かるように、単純なUDPと比較しても、RtMPのプロトコルオーバーヘッドは同程度であり、十分な性能を出していることがわかる。

【0094】次に、実時間マルチキャストプロトコルに必要とされる機構について、RtMPと他のプロトコル（ST-11、RSVP（Resource ReSerVation Protocol））との比較を行い、特に、RtMPの特徴である受信者からの経路選択機構の有効性、および既設接続の保証について検討する。

【0095】最初に、接続ノードの柔軟性について説明する。まず、送信者および受信者から指定された要求に対する適応度について考察する。各接続のFlowspecは、各受信者の要求、および利用可能な資源の量によって定められる。送信者から送られたデータは、指定された優先度レベルにより転送可能性が判断され、転送可能な接続にのみ送られる。

【0096】従って、受信者の要求が高く、経路上で利用可能な資源が多いホストでは高いQOS（Quality Of Service）のデータを、また、受信者の要求が低いか、

または利用可能な資源が乏しい場合は、低いQOSのデータをそれぞれ受け取ることができ、多種多様な受信者の要求、および各経路の特性に対応することができる。

【0097】次に、RtMPの最大の特徴である経路制御部と、資源予約プロトコルとの統合の効果について説明する。RtMPにおいては、経路の資源予約に関するパラメータのみをRtMP側で管理し、受信者からの要求に応じて経路選択を行うことにより、経路制御アルゴリズムに依存することなく、送信者および受信者の要求に適応したツリーの構築を可能としている。ST-11やRSVPにおいては、資源予約機構は経路制御とは独立に設計されているため、受信者の要求に応じた経路制御は困難である。

【0098】さらに、他のコネクション設立と競合するため、経路制御部によって提案された経路が要求を満たすことができない場合には、他の経路を選択することによって、再度コネクションの設立を試みる。これにより、コネクション設立の可能性はさらに向上する。

【0099】これらのことから、RtMPは、既存の実時間マルチキャストプロトコルと比較して、より送信者および受信者からの要求に応じた柔軟性の高いコネクション設立機構を提供するということができる。

【0100】次に、既設コネクションの保証について説明する。資源予約方式の実時間通信プロトコルは、実時間性を満たすことを最大の目的とし、そのために予めコネクションを設立し、必要な資源を予約する。従って、一度設立されたコネクションは、ホストおよびネットワークの障害発生時を除き、コネクション設立時の要求を常に満たすことを保証しなければならない。

【0101】1対多通信においては、ST-11およびRtMPにおいて、コネクション設立時に、新たなコネクションの設立が既設のコネクションを乱さないことを確認する。その後、コネクションが変更されることはないため、既設コネクション上のデータ送信は、常に要求を満たすことが保証される。

【0102】多対多通信においては、重要な特性の1つとして、グループの性質を考慮した複数の送信者によるコネクションの共有が挙げられ、RSVPおよびRtMPはこの機構を提供する。ここで、複数の送信者が1つのコネクションを同時に使用しようとした場合、転送データのオーバーフローが生じ、双方ともデータ転送を保証することは不可能となる。

【0103】RtMPにおいては、コネクションの共有などの性質を送信者が決定し、送信者リストにより送信者間で協調し合うことを可能としているため、転送データのオーバーフローの発生を防ぐことができる。これに対して、RSVPにおいては、コネクションの共有は受信者によって決められ、送信者側からの保証がなされない。従って、多対多通信においても、RtMPはRSVPと比較して既設コネクションをより保証していると言

える。

【0104】次に、プロトコルの耐障害性について説明する。RtMPおよびST-11は、隣接ノードを監視することにより、障害発生を検知および障害対策を行う。RSVPにおいては、障害の発生をプロトコル自身で監視するのではなく、独立した経路制御アルゴリズムによって監視され、障害の発生が検知されると、経路アルゴリズムによって経路変更が行われ、自動的にツリー構造が変更されるようになされている。このことから、障害に対する柔軟性という点においては、RSVPは非常に柔軟であるということが出来る。

【0105】しかしながら、実時間通信という点を考慮に入れた場合、障害の発生によるデータ転送の停止は致命的である。従って、他の機構によって自動的に経路が切り替えられるのではなく、実時間通信プロトコルによってユーザへの通知および適切な処理が行われる必要がある。このような処理を行う場合においては、RSVPより、ST-11やRtMPの方が適しているといえる。

【0106】以上、従来分離されていたマルチキャスト経路制御機構とコネクション設立機構の統合、およびそれに基づいた実時間マルチキャストプロトコルRtMPについて述べた。上述したように、RtMPは、受信者からの要求を経路制御に反映することにより、多種多様な受信者の要求に対応することが可能であり、また、送信者リストの管理機構により効率がよく、かつサービスレベルを保証する多対多通信を提供することができる。

【0107】なお、上記実施例においては、RtMPをRT-Mach上で実装した場合について説明したが、これに限定されるものではない。

【0108】また、本発明は、ATM(Asynchronous Transfer Mode)などの新しいネットワークアーキテクチャに適用することも可能である。

【0109】

【発明の効果】請求項1に記載の通信制御方法によれば、受信者からの要求に基づいて、送信者と受信者との間の経路を選択し、コネクションを設立する経路選択機構と、経路の資源を予約する資源予約機構の間のインタフェースを規定し、経路選択機構と資源予約機構とを統合するようにしたので、1対多通信において、受信者の要求に応じた経路を選択し、コネクションを設立することができ、既設のコネクションの品質を保証することが可能となる。

【0110】請求項3に記載の通信制御方法によれば、各送信者が保持する各送信者のリストに基づいて、受信者から各送信者に対してコネクションの設立を行い、複数の送信者がコネクションを共有する場合、コネクションに予約された資源の量に基づいて、コネクションを介してデータを受信者に同時に送信する送信者の数を所定数以下に制限し、新たな送信者または受信者のネットワ

ークへの参加要求は、新たな送信者または受信者がネットワークに参加した場合においても、既存のコネクションを設立している送信者および受信者のQOS要求が保証されるとき、受理されるようにしたので、多対多通信を実現し、かつ、既設のコネクションの品質を乱さないことを保証することが可能となる。

【図面の簡単な説明】

【図1】経路制御部と資源予約機構間のインタフェースの例を示す図である。

【図2】互いに異なる転送能力の経路が分岐する中間ノードの例を示す図である。

【図3】優先度レベルとデータ転送レートの対応関係の例を示す図である。

【図4】コネクション設立手順の例を説明するための図である。

【図5】障害発生時の処理手順の例を説明するためのフローチャートである。

【図6】送信者参加時のコネクション設立手順の例を説明するためのフローチャートである。

【図7】受信者参加時のコネクション設立手順の例を説明するためのフローチャートである。

【図8】RTMPの内部構成例を示す図である。

【図9】受信者および送信者の参加に要する時間を示す図である。

\*

\*【図10】ラウンドトリップ時間とメッセージサイズの関係を示すグラフである。

【図11】従来のプロトコルST-IIとRSVPの特徴を示す図である。

【符号の説明】

1乃至6 ホスト, 11 アウトプットスレッド (Output thread), 12 ローカルセンドスレッド (Local send thread), 13 フォワーダスレッド (Forwarder thread), 14 コントロールスレッド (Control thread), 15 ネットワークデバイスレイヤ (Network device layer), 16 プロトコルレイヤ (Protocol Layer), 17 インタフェースレイヤ (Interface Layer), 18 コネクションテーブル (Connection table), 19 イーサネットスレッド (Ether input thread), 20 ネットワークプロトコルサーバ (Network Protocol Server), 21 ルーティングサーバ (Routing Server), 22 アプリケーション (Applications), 23 RT-Machマイクロカーネル (RT-Mach Micro Kernel)

【図1】

Interface 名	引数	返り値	動作
RouteAdd	経路 (A,B) 経路 (B,C)	経路 (A,C)	経路 (A,C) の資源 の量を計算する
RouteSelect	経路リスト	経路リスト	必要な経路を 選択する
RouteEval	経路	評価結果	経路の重み付け を行う
RouteReq	送信者	経路リスト	上位ノードの候補 を傳る

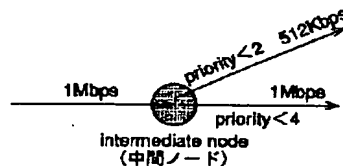
【図3】

優先度レベル	データ転送レート	内容
1	192 Kbps	音声
2	512 Kbps	粗い画像
3	1 Mbps	細かい画像

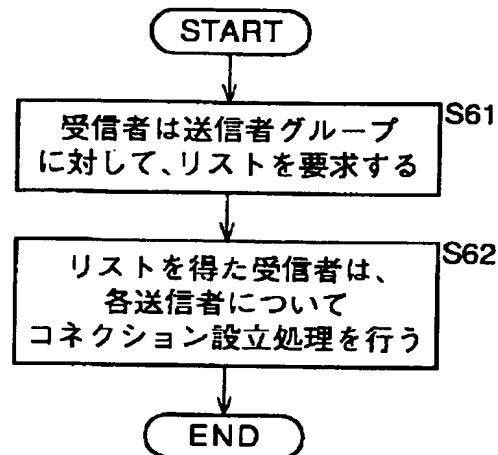
【図9】

	In same segment	through one gateway
Join as receiver	22.4 msec	31.4 msec
Join as sender	23.4 msec	40.4 msec

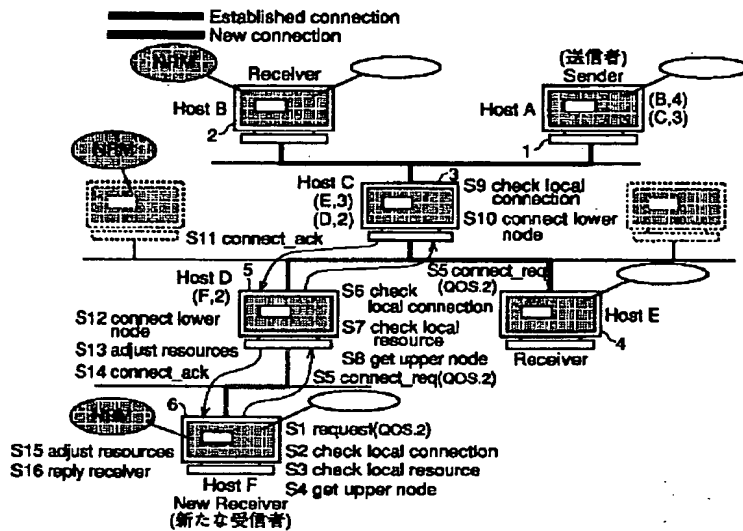
【図2】



【図7】



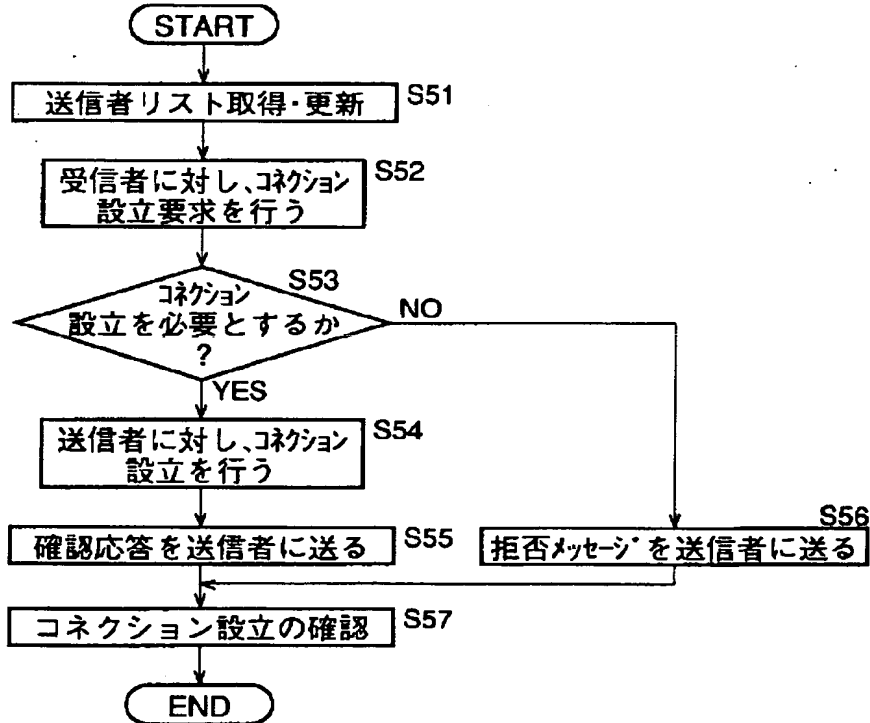
【図4】



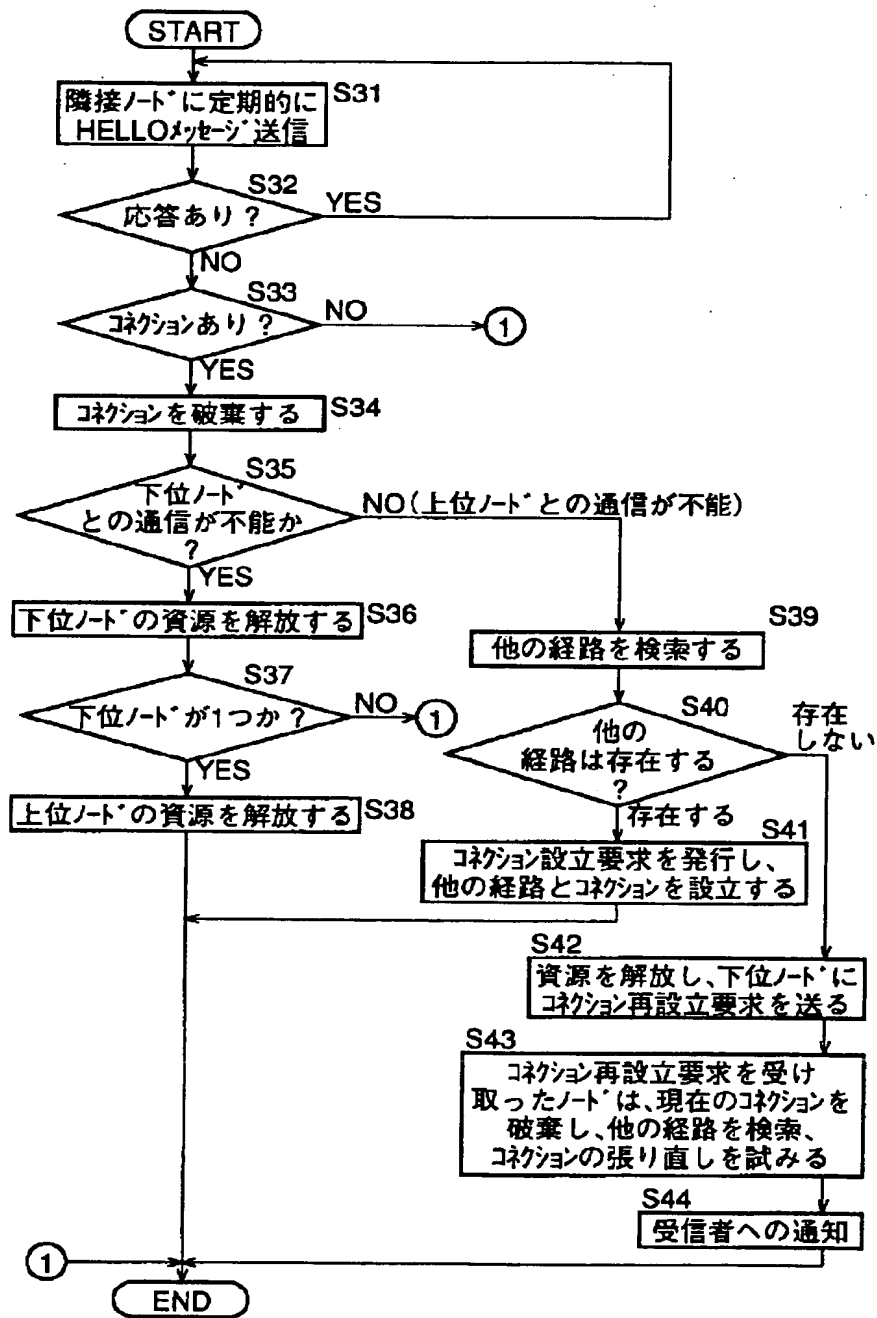
【図11】

	ST-II	RSVP
経路選択	送信者	送信者
コネクション設立	送信者	受信者
通信経路	静的	動的
スケーラビリティ	×	○
既設コネクションの保証	○	×
各受信者の要求	×	△

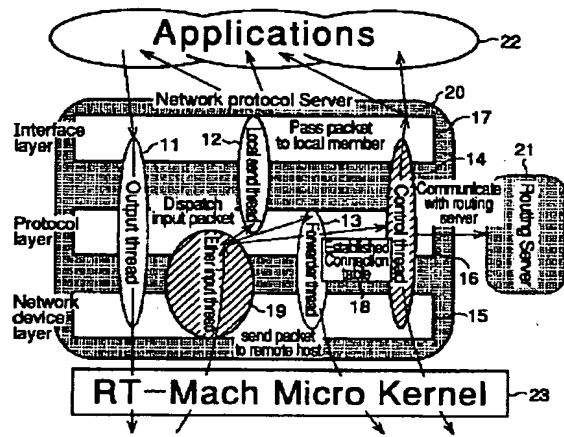
【図6】



【図5】



【図8】



【図10】

